



US005548506A

United States Patent [19]
Srinivasan

[11] **Patent Number:** **5,548,506**
 [45] **Date of Patent:** **Aug. 20, 1996**

[54] **AUTOMATED, ELECTRONIC NETWORK
 BASED, PROJECT MANAGEMENT SERVER
 SYSTEM, FOR MANAGING MULTIPLE
 WORK-GROUPS**

[76] **Inventor:** Seshan R. Srinivasan, 1524 Condor
 Way, Sunnyvale, Calif. 94087

[21] **Appl. No.:** 210,172

[22] **Filed:** Mar. 17, 1994

[51] **Int. Cl.⁶** G06F 17/50; G06F 17/60;
 G06F 15/163

[52] **U.S. Cl.** 364/401 R; 364/226.7;
 364/468.01; 395/200.01; 395/200.12; 395/200.15;
 395/919; 395/923; 395/926

[58] **Field of Search** 364/401; 395/600,
 395/200.01, 200.03, 200.06-200.18, 919,
 923, 926

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,451,067	5/1984	Williams	281/31
4,875,162	10/1989	Ferriter et al.	364/401
5,168,444	12/1992	Cukor et al.	364/401
5,255,181	10/1993	Chapman et al.	364/401
5,381,332	1/1995	Wood	364/401

OTHER PUBLICATIONS

"Project Management Software That's IS Friendly", Lee, Datamation, p. 55(4) Oct. 1993.

"Syzygy: Aiding a stellar of resources" by Joseph Devlin, Personal Computing, v. 14, n4, p. 152(1) Apr. 1990.

"Groupware rocks the enterprise", by Mitch Irsfeld, Open Systems Today, p. 32(1) Aug. 1993.

"Primavera Project Planner: Project Management Software", PC Magazine, p. 170(1).

"High-End Project Manager: Coordinate Enterprizewide Project With Desktop Flexibility" by Heck, Mike: InfoWorld, V15, n5, p. 59(9); Feb. 1, 1993.

Primary Examiner—Thomas G. Black

Assistant Examiner—Hosain T. Alam

[57] **ABSTRACT**

Design and implementation of an 'Auto Multi-Project Server System', which automates the tasks of Project Management Coordination, for organizational work-group team members. The 'Auto Multi-Project Server', referred to as AMPS, consists of a core piece of software running on a host server computer system and interacting with a messaging system such as electronic mail, fax etc. Once the AMPS system is configured for the work environment, all interactions with it by work-group team members is via messages. First the AMPS system compiles multi-project plans into a multi-project database, and tracks the ownership of projects, tasks and resources within the plans. Second the AMPS system performs automatic checking of resource requests, if resource availability limits are exceeded then resources are re-allocated to projects based on priorities, and project plans are accordingly changed Third the database is processed periodically to send out reminder follow-ups and project status reports. Fourth the databases are continuously updated based on status changes reported by work-group members. These four steps are continuously repeated enabling an automated method of multi-project management for organizational work-group team members.

9 Claims, 10 Drawing Sheets

The 'Auto Multi-Project Management' Process

Objective: Architecture drawing to show the different components for implementing the automatic multi-project server, to use in an automatic multi-project management process.

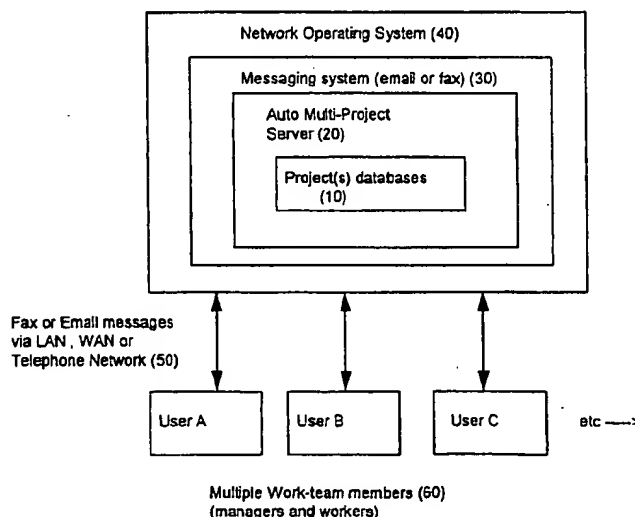


Fig 1. The 'Auto Multi-Project Management' Process

Objective: Architecture drawing to show the different components for implementing the automatic multi-project server, to use in an automatic multi-project management process.

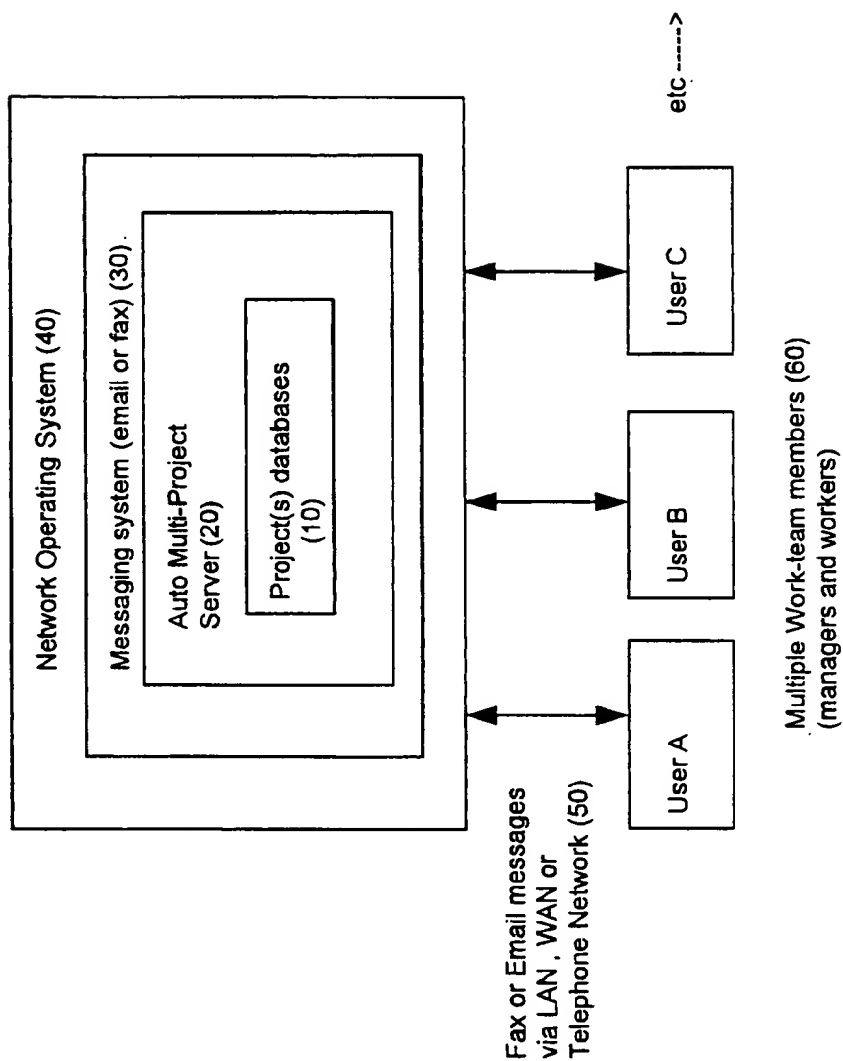


Fig 2 - Internal Architecture of the 'Auto Multi-Project Server'

Objective: Show the internal components of the 'Auto Multi-Project Server'.

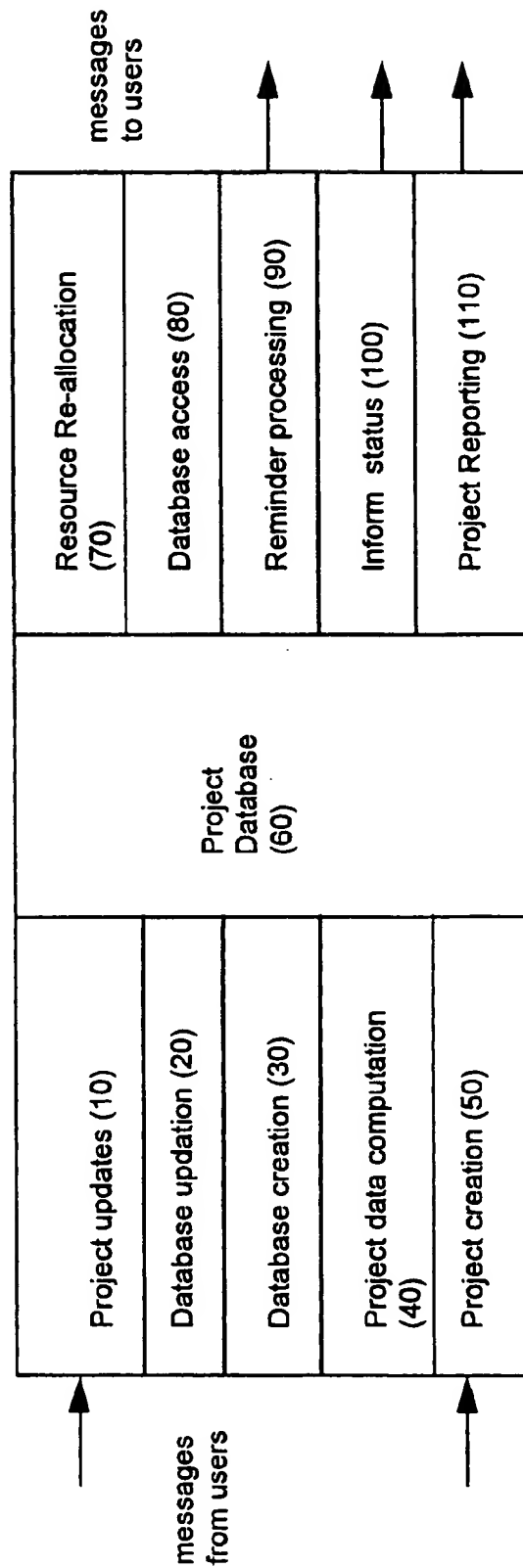


Fig 3 - Proj ct database example

User Name > xxx
Password > 12345678

Project Description (1 line) > xxxxxx
Project Id No (upto 3 characters) > 3456
Project Start Date (m/d/y) > 2/20/1994
Project Leader Name > Sesh Srinivasan
(optional) Project Leader Email Id > sesh@xyz.abc.com
(optional) Default Reminder Frequency (d,b,w,f,m) >
(Optional) Week-ends (sat,sun,mon,tue,wed,thu,fri) > sat sun
(optional) Holidays (m/d/y) >

Task Id (numbers only) > 1
Task Description (1 line) > procure pipes
Task Status (pending/started/completed/stuck) > started
Task Leader Name > tl1
Task Leader Email Id > tl1@xxx.yyy.com
Task Duration (d,w,m or 0) > 3d
OR Task Completion Date (m/d/y) >
(optional) Task Reminder Frequency (d,b,w,f,m) >
(optional) Task Leader Can Change Schedule (y/n) > y
(optional) Parent Task Id >

Immediate Predecessor Task Id >
Dependency Type (fs,ff,ss) >
Time Lag (d,w,m) >

Resource name (include units) > Feet of copper pipes
Resource budget (numbers only) > 2000
Resource consumed > 500

Task Id (numbers only) > 2
Task Description (1 line) > install pipes
Task Status (pending/started/completed/stuck) > pending
Task Leader Name > tl2
Task Leader Email Id > tl2@abc.pqr.com
Task Duration (d,w,m or 0) > 1w
OR Task Completion Date (m/d/y) >
(optional) Task Reminder Frequency (d,b,w,f,m) >
(optional) Task Leader Can Change Schedule (y/n) > y
(optional) Parent Task Id >

Immediate Predecessor Task Id > 1
Dependency Type (fs,ff,ss) > fs
Time Lag (d,w,m) > 1d

Resource name (include units) > plumber man days
Resource budget (numbers only) > 10
Resource consumed >

Fig 4 - 'Build' Program Flowchart

Objective: To compile together data on inter-dependent tasks for each project and build a reference database
 To compute completion dates based on resource limits and inter-project dependencies.

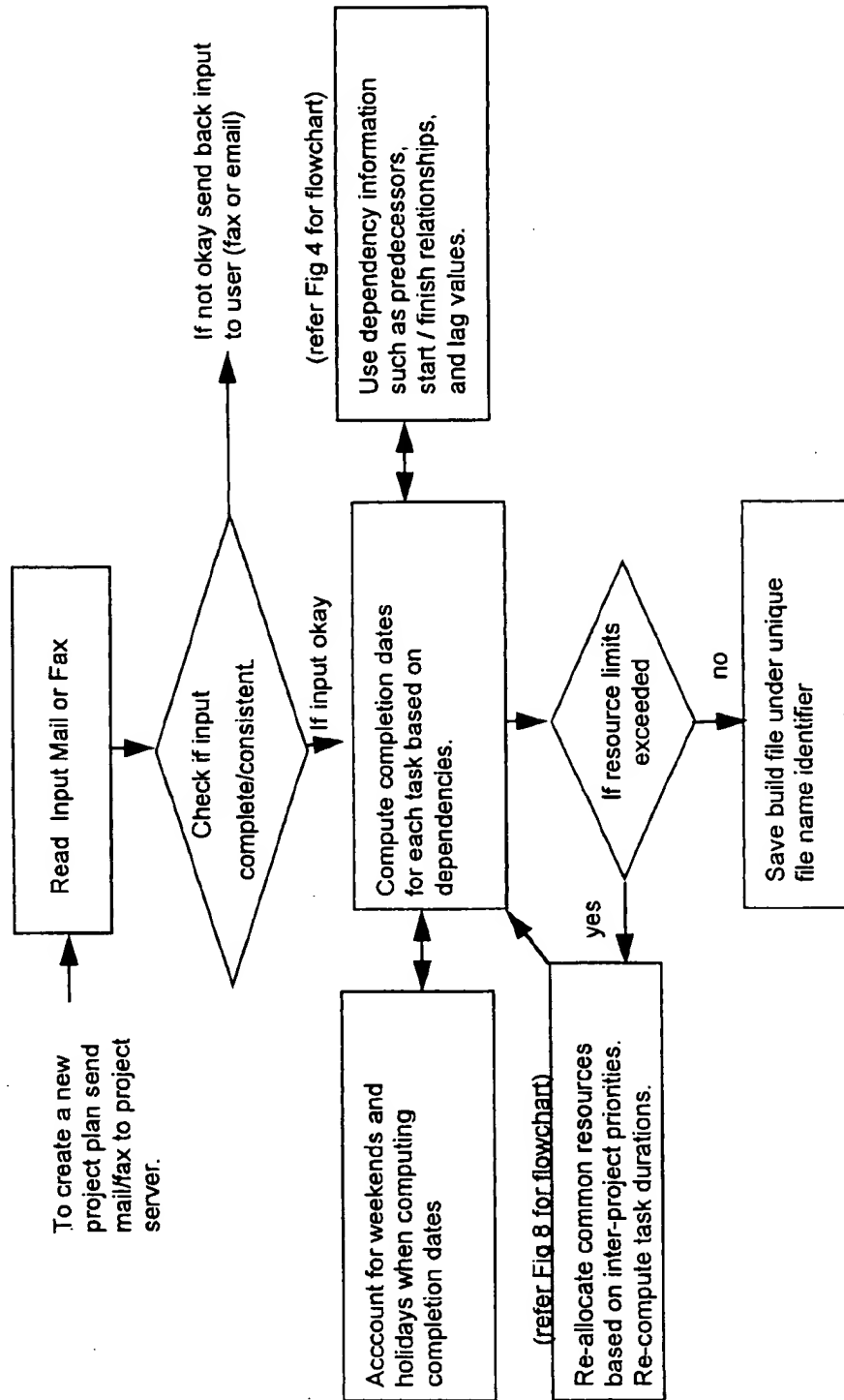


Fig 5 - Dependency Computation Flowchart

Objective: To compute task start and finish dates, based on relationship to dependent predecessors.

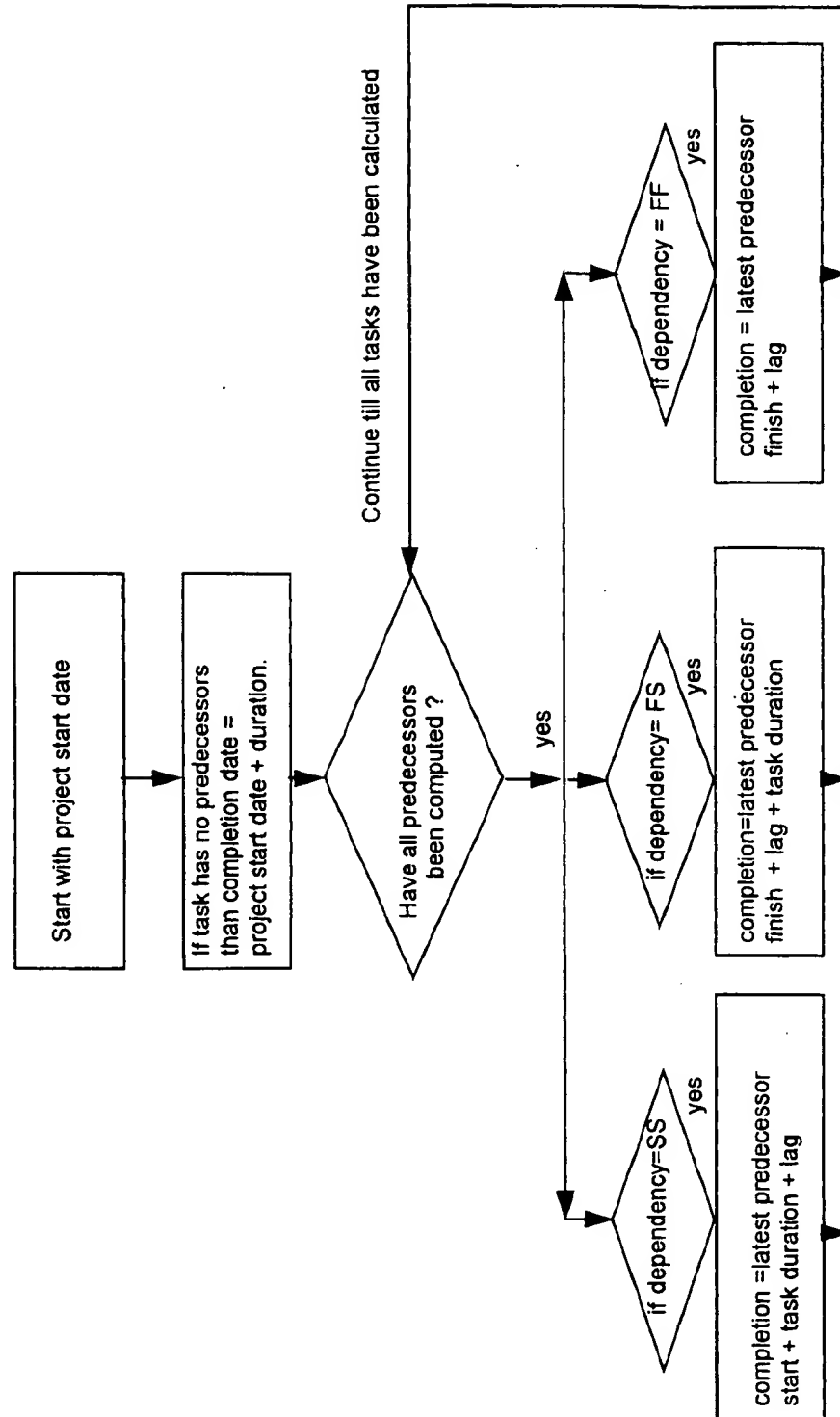


Fig 6 - 'Reminder' Program Flowchart

Objective: To track pending tasks and remind task owners on a pre-determined frequency on when to start/finish their tasks.
To update task owners on status of dependencies.

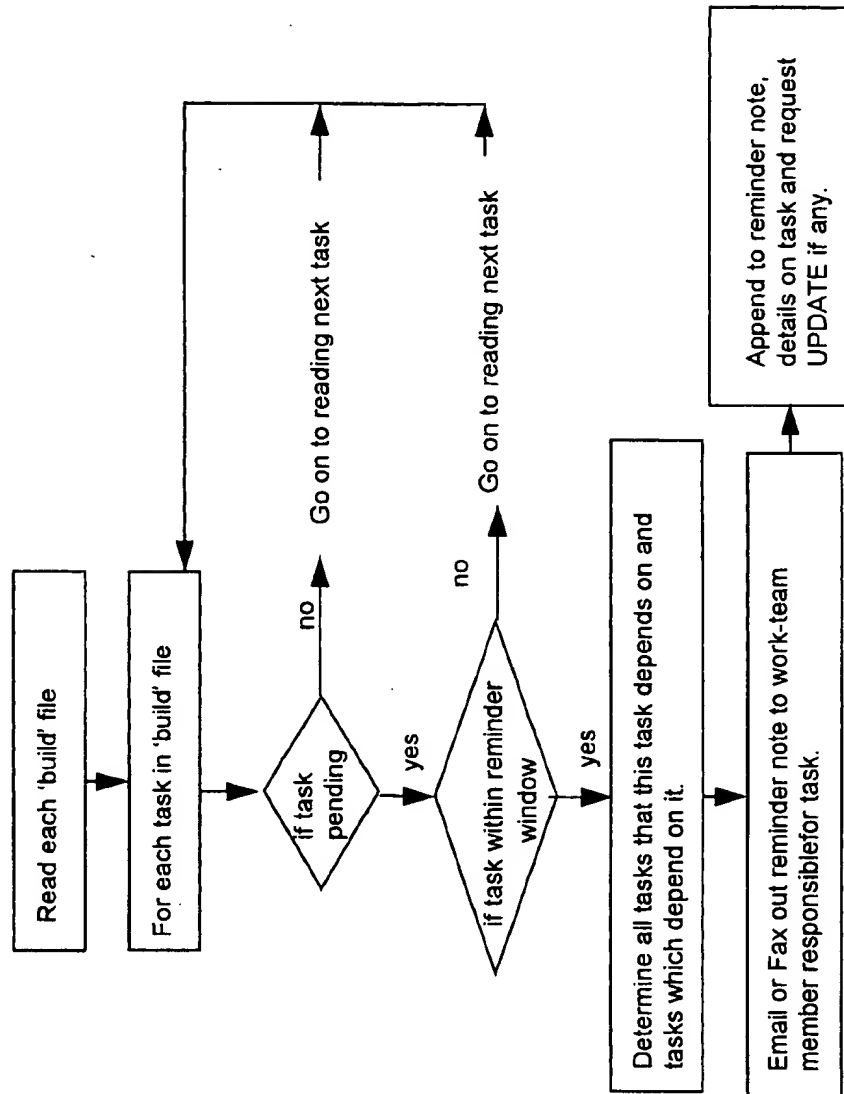


Fig 7 - 'Inform' Flowchart

Objective : To track task completions and inform dependent tasks on start or finish status of their predecessors, thus to act as a formal channel of communication eliminating the chance of mis-information.
To inform project and task leaders on re-computed completion dates based on resource reallocation between projects.

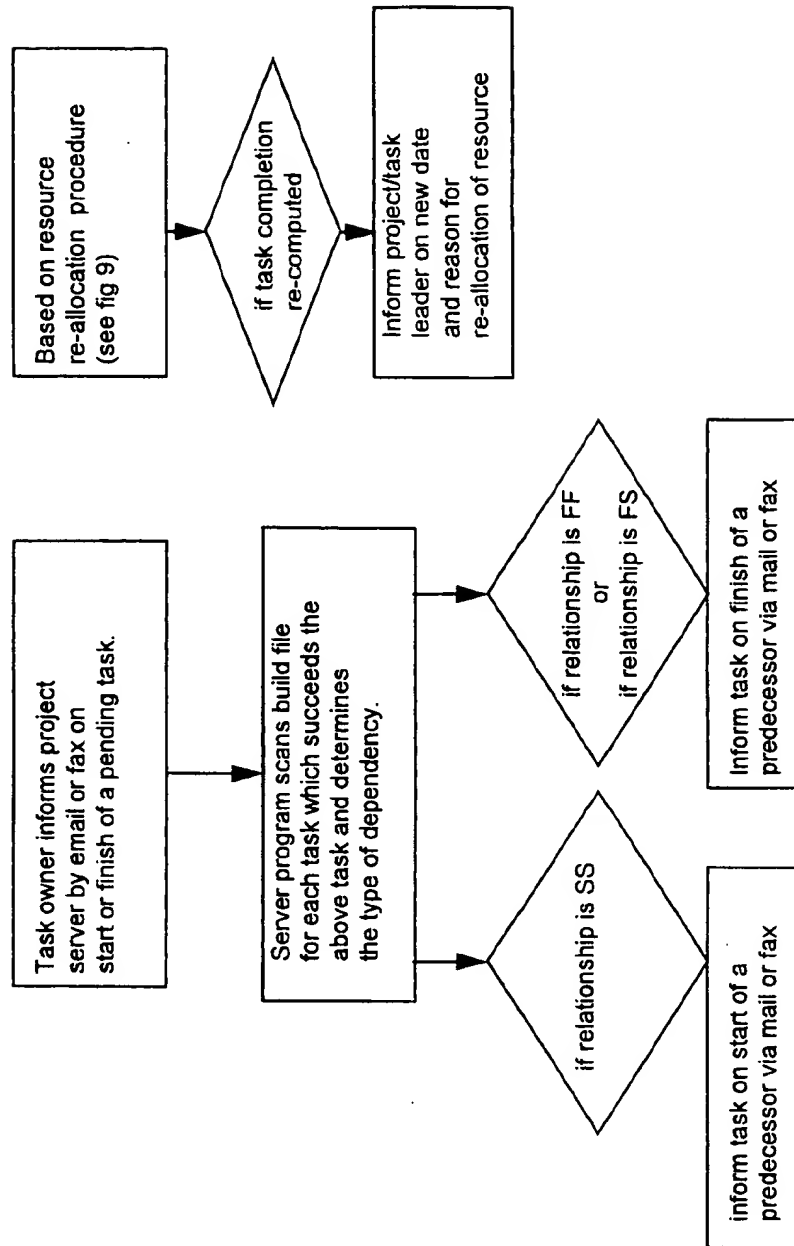


Fig 8 - 'Update' Flowchart

Objective: To update the 'build' file with status changes in task details, or task inter-dependencies or inter-project priorities.

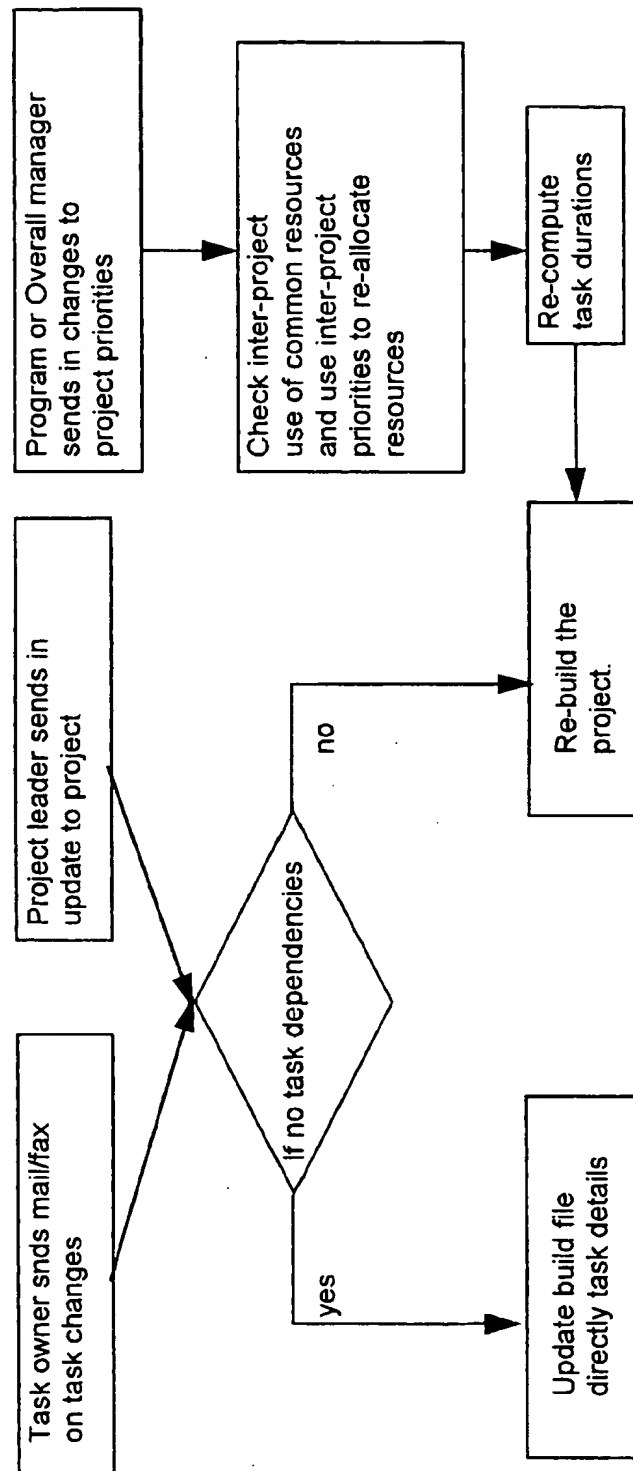


Fig 9 - Inter-project priorities and resource allocation flowchart

Objective: To allocate common resources based on inter-project priorities and re-compute task durations.

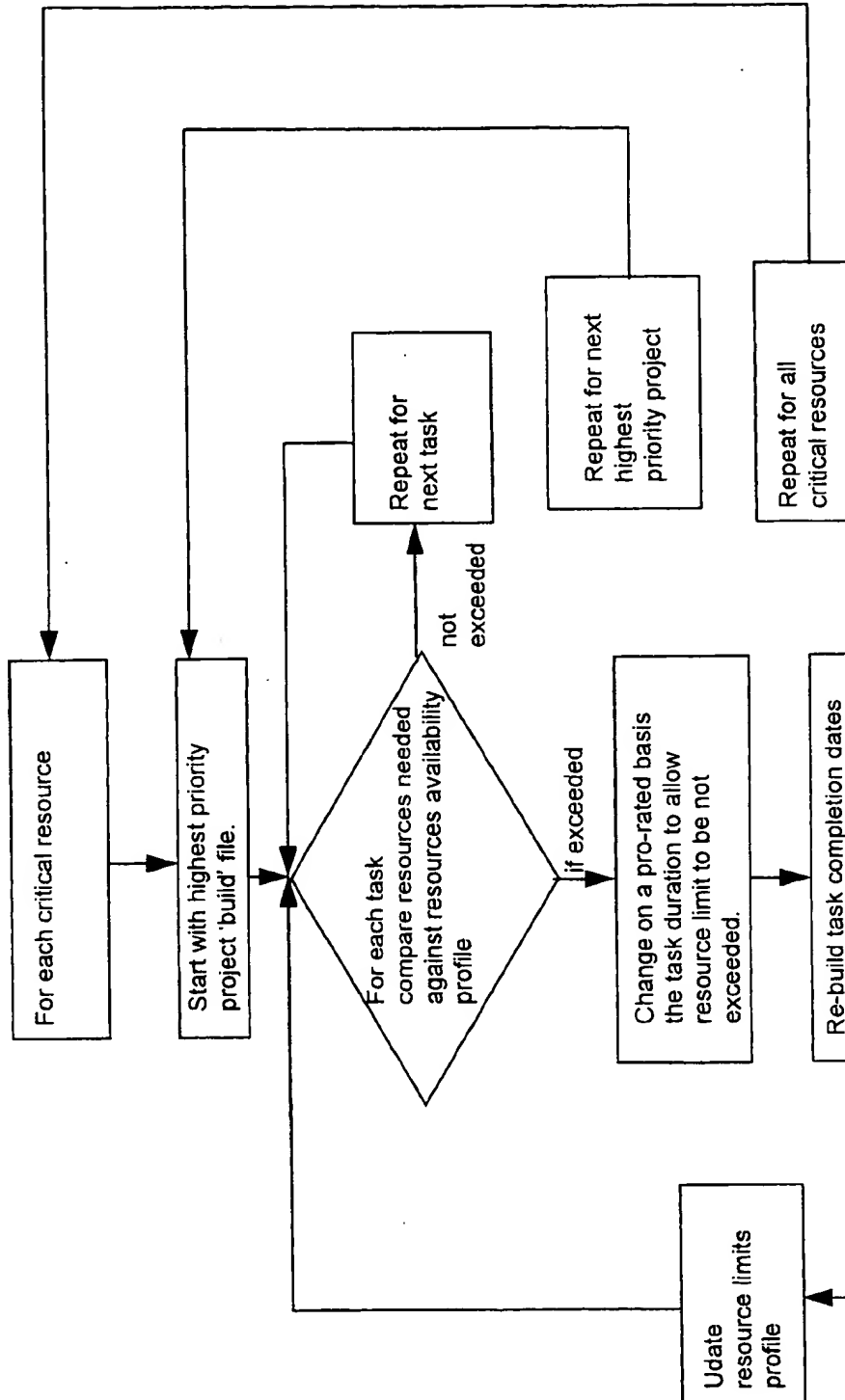
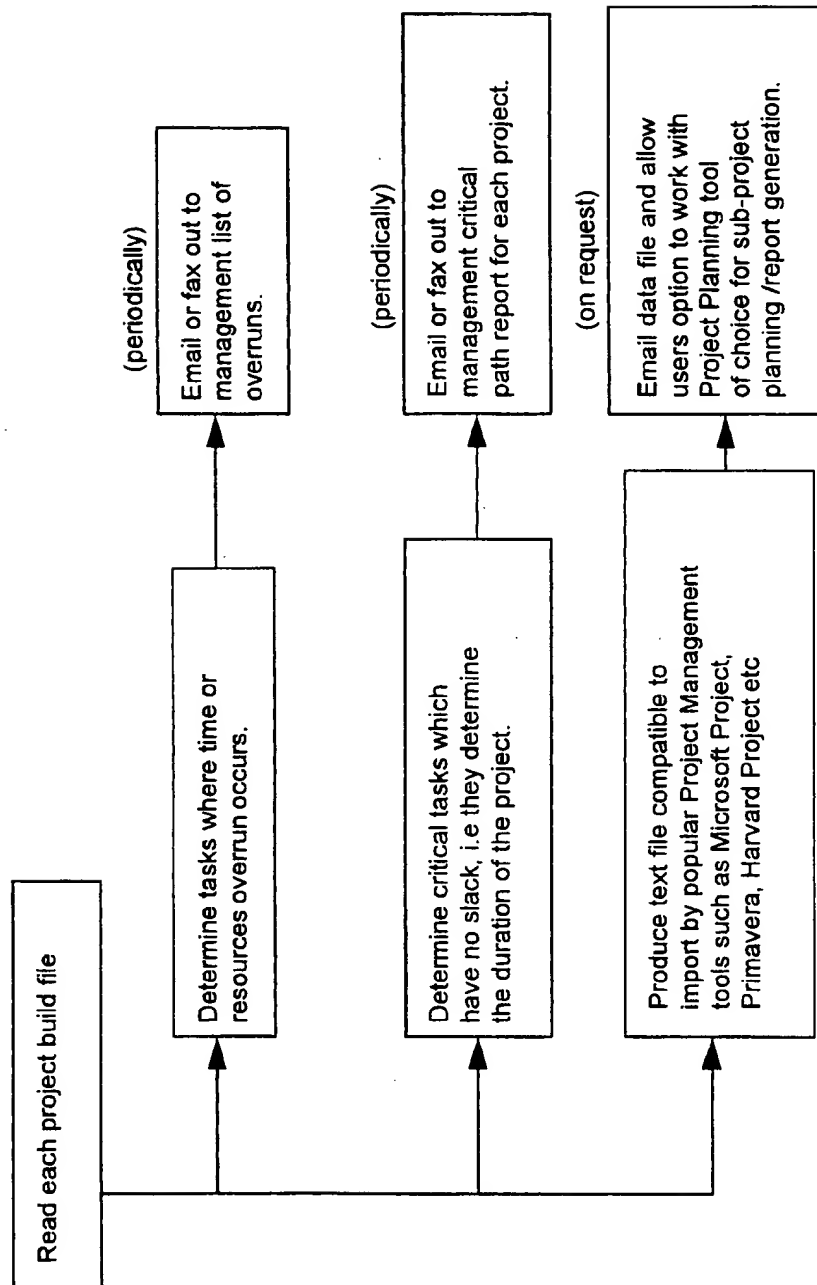


Fig 10 - 'Reports' Flowchart

Objective: To generate project management reports for reporting progress of project(s).



AUTOMATED, ELECTRONIC NETWORK BASED, PROJECT MANAGEMENT SERVER SYSTEM, FOR MANAGING MULTIPLE WORK-GROUPS

BACKGROUND

1. Cross-references to Related Applications

A patent search revealed no patent under the categories, Automatic Project Management, or under Electronic Mail based ~~Project Management or Groupware Project Management.~~

The category Project Management revealed two patents:

- (1) U.S. Pat. No. 4,451,067, A Comprehensive Central Scheduling Folder for Project Management offers a non-computerized method for planning and tracking projects. This has no similarity with the proposed electronic network based project management system, as it does not involve computers.
- (2) U.S. Pat. No. 4,875,162, An Automated Interface of Design Software with Project Management Software. The scope of this is narrow, focuses on design organizations only and involves automatic status update of files between design software and project Management software. This is not for general purpose management of multiple work-groups. Also it does not address the design of a general purpose auto-server for project management, which will perform automatic project database creation, project follow-up, multi-project resource planning and project reporting.

2. Field of Invention

The invention described here is the design and application of an electronic network based project management server system. The product termed in this application as an 'AUTO MULTI-PROJECT SERVER'. The use of the product will result in an improved organizational process for compiling, tracking and managing multiple projects within an organization.

Program
The goal of the 'Auto Multi-Project Server' is to act as an automated computer based project coordinator to manage the goals of multiple organizational work teams. Activities of the automated computer based server comprise of collating/compiling project data, flagging inconsistencies, follow-up with work team members, obtain updated project tracking data, communicate project progress to work team, resolve inter-project conflicts by re-allocation of critical resources based on project priorities and generate management reports for flagging time and cost overruns and critical path information.

Future re-engineered corporations will demand heavy computing needs for managing large number of work-teams with different project goals, schedules, dependencies and priorities running simultaneously. The present techniques for static project-management tools and for limited project-management groupware technologies will no longer be adequate.

3. Discussion of Prior Art

Presently the following types of project management systems are available:

1. Project planning tools, with software for generating charts such as Gantt, Pert/CPM etc. Many of the techniques were invented during the Second World War Majority of project management is perceived as belonging to this area. These are static standalone tools and do not impose any particular process in the organization. Organizations have to figure out methods for input, tracking and report distribu-

tion. Typically a project coordinator is hired by organizations to manually compile input data, run the tools and distribute progress reports.

Examples: Commercial tools such as PRIMAVERA and MICROSOFT-PROJECT offer computerized static software approach to planning projects, i.e. a person has to supervise the input data collection and output report generation, and there is no automatic process specified by the manufacturer.

A survey of existing project management tools showed no use of electronic mail or faxing methods to do any automatic creation, updating, planning or reporting. (Reference: DATAMATION—Oct. 1, 1993, Article: Project Management Software that's IS friendly, has a survey of the capabilities of Project Management tools and there is no mention of any automatic server based tools, driven by electronic messaging.)

2. Group collaboration and Work flow management tools, provide technologies to automate the flow of documents. These are directed at a wide variety of needs such as facilitating group conferences, meetings' management, project database management, information flow control and other custom mail enabled applications. The tools by themselves do not impose any organizational process, but offer an underlying technology to build applications, including the design proposed by this patent.

Example (1): U.S. Pat. No. 5,168,444 for a Work-flow shipment system, including processing of document images. This is not related to automatic project management for multiple work-groups.

Example (2) Lotus NOTES is a leading groupware product offering the necessary technology to build organizational processes including Project Tracking examples. The system is not designed to handle large multi-project/multi-priority environments. There is no automatic system proposed by the manufacturer and it is left for each organization to design its own process or system of project management.

Example (3) The nearest commercial product design is a product called SYZYGY from Information Research Corp., which builds on inherent capability of local area networks (LAN) to tie workers together. The system has a central database on organizational projects and can be automatically be programmed by managers to broadcast reminder messages. (Reference: Personal Computing, Volume 14, Issue 4, Date: Apr. 27, 1990, Page: 152), The differences I see between this product and the invention under this patent:

Syzygy is not ideal for managing large or complex projects since it lacks critical path analysis. The invention under this patent includes a complete computation engine for computing critical path activities.

Syzygy does not do complex inter-task dependency computations.

Syzygy is not targeted for multiple work-teams with inter-conflicting priorities.

Syzygy does not attempt any resources leveling or resource re-allocation.

Objects and Advantages

The product in this patent application the 'Auto Multi-Project Server' consists of a centralized automatic project management server software, based in a computer and accessible via electronic mail or fax messages.

No Programming is involved by people using the system, the 'Auto Multi-Project Server' automatically computes task completion/start dates and triggers messages based on data mailed to it. Thus the data itself is used to program the 'Auto Multi-Project Server'.

3

The 'Auto Multi-Project Server' reads electronic mail or fax messages, such as project plans and flags project data inconsistencies. It then compiles project management data into a usable project database with information to include details on goals, milestones, budgets etc. The 'Auto Multi-Project Server' follows-up with task leaders on pending tasks by reminding them on task start and finish dates.

The 'Auto Multi-Project Server' includes extensive dependency calculations, such as taking into account complex inter-task dependencies and start/finish lags between dependent activities.

The 'Auto Multi-Project Server' goes beyond just reminding people about their start and finish dates, it also serves as a project information system by informing the group team members on their inter-dependencies.

The 'Auto Multi-Project Server' includes resolution of multi-project priorities, which is an essential component for managing multi-groups with divergent priorities. It allows for example a program or general manager to examine multiple-projects under his program and assign priorities for completion. Individual task deadlines and resource usage would be controlled by the priorities assigned to the completion of the project.

The 'Auto Multi-Project Server' will re-allocate critical resources based on project priorities and re-compute completion dates. The project database would automatically be updated with the new dates. Thus there is automatic resolution of priority conflicts. The project leader does not have to have to constantly escalate issues up the management hierarchy for resolution.

Security issues are handled by the authorization of all input mail or faxes, prior to processing. For electronic mail and faxes the incoming mail is checked against a valid list of passwords prior to processing. Also each project is assigned a unique identifier to prevent mix-up of information between projects.

In summary the 'Auto Multi-Project Server' is a necessary requirement for the needs of future HORIZONTAL CORPORATIONS where the information to manage multiple projects will be too complex to be done via manual present-day tools.

Further objects and advantages of my invention will become apparent from a consideration of the drawings and ensuing description.

DESCRIPTION OF DRAWINGS

FIG. 1—Architecture drawing to show the different components of an 'Auto Multi-Project Management Process', which is required for implementing the 'Auto Multi-Project Server'.

FIG. 2—Internal architecture of the 'Auto Multi-Project Server'.

FIG. 3—Project database example, format to serve as the primary communication and storage format for information on the different project(s).

FIG. 4—Build Program Flowchart, to compile together data on inter-dependent tasks for each project and build a reference database (with all necessary project information). To compute completion dates based on resource limits and inter-project dependencies.

FIG. 5—Dependency computation flowchart, to compute task start and finish dates based on relationship to dependent predecessors.

4

FIG. 6—Reminder program flowchart, to track pending tasks and remind task owners on a pre-determined frequency on when to start/finish tasks. To update task owners on status of dependencies.

FIG. 7—Inform flowchart, to track task completion and inform dependent tasks on start or finish status of their predecessors, thus acting as a formal channel of communication eliminating the chance of misinformation. Also to inform project and task leaders on re-computed completion dates based on resource re-allocation between projects (based on project priorities).

FIG. 8—Update flowchart, to update the 'build' file with status changes in task details or task inter-dependencies or inter-project priorities.

FIG. 9—Inter-project priorities and resource allocation flowchart. To allocate common resources based on inter-project priorities and re-compute task duration.

FIG. 10—Reports flowchart, to generate project management reports for reporting progress of projects(s).

LIST OF REFERENCE NUMERALS

FIG. 1

- 10 Project(s) databases, contains information on the project, also referred to as the 'BUILD' file. Format for this file is shown in FIG. 3.
- 20 Auto project management server. This software system is the heart of the process and its workings are explained in the flowcharts contained in FIG. 4 through FIG. 10.
- 30 Messaging system such as Fax or Email (Electronic Mail)
- 40 Network Operating System, is a general term used to refer to the communication protocol used to allow users communicate with the 'Auto Project Server'.
- 50 Fax, LAN (local area network), WAN (wide area network) or telephone network is the media for communication.
- 60 Work-team members communicating to the 'Auto Multi-Project Server', these may be program managers, project leaders, task leaders etc.

FIG. 2

- 10 Project updates, module to parse and read project updates,
- 20 Database updating, module to update the project database with new information.
- 30 Database creation, module to open a new database project file.
- 40 Project data computation, module to compute all project data, (Dates etc.)
- 50 Project creation, module to parse and read in project data on new projects.
- 60 Project database, centralized complete information on the project.
- 70 Resource re-allocation, re-assigns critical resources if limit exceeded.
- 80 Database access, to retrieve task information from project database.
- 90 Reminder processing, to send out reminders to pending task owners.

5

100 Inform status, to communicate status of dependencies.

110 Project reporting, to generate and send out project progress reports.

SUMMARY

The design and implementation of an 'Auto Multi-Project Server' for Automatic Multiple Project Management is described in this patent. This involves a self-running software system running on a central server computer system with capabilities for automatic data compilation, tracking and management, handle multiple projects, resolve inter-project resource conflicts and communicate with users via electronic mail or fax mail.

DESCRIPTION OF INVENTION

FIG. 1 shows the overall process of using the invention in an organization to manage multiple projects via an automatic computer server. FIG. 2 shows the internal architecture of the invention which is an 'Auto Multi-Project Server'.

FIG. 1, Part 10—refers to the set of project database files stored, one for each project. These files are also referred to as the 'build' file. Refer FIG. 3 for a sample format. This file contains information about the project, such as the start date and other global information such as the project leader's name, mail identity, holidays, weekends etc. It also contains detailed information about each task such as the description, task leader's name/mail identity, task duration or planned completion date. In addition it contains information about the dependencies for each task, in terms of the predecessor task id, the type of dependency whether FS(predecessor finishes and task starts), FF(predecessor and task finish together) or SS (predecessor and task start together), and the lag between the predecessor and task activities. Finally the database contains information about which resources, amount of resources and budget for resources required for doing the task.

FIG. 1, Part 20 is the 'Auto Multiple Project Server'. This is a software system which is the head of the new process. Details of its internal architecture are contained in FIG. 2. Its workings are explained in the flowcharts contained in FIG. 4 through FIG. 10. The server is a software process that runs at fixed intervals (example: at the end of the day) and performs the task of project management coordination by performing date compilation, data validation, database update, follow-up reminders to task leaders, inter-project resource conflict resolution and allocation and management reports generation on critical tasks and overruns. The main modules of the server software are described below:

Creation functions consisting of:

Project creation module to read and check for consistency, project data on new projects. (Component 50, FIG. 2)

Data computation module, to compute completion dates for all activities in the project based on dependency information and project start date. (Component 40, FIG. 2)

Database creation, this module creates a new database project file. (Component 30, FIG. 2).

Project database, holds centralized complete information on the project. (Component 60, FIG. 2).

Conflict-resolution functions consisting of:

'Resource' module, to check for resource usage and to re-allocate resources on the basis of inter-project priorities, if resource are limits exceeded. Project database

6

has to be re-created if resources are re-allocated. (Component 70, FIG. 2).

Updating functions consisting of:

'Project Update' module, to receive and parse updates from project/task leaders (Component 10, FIG. 2)

Database updating, module to update the project database with new project status information. (Component 20, FIG. 2).

Information functions consisting of:

Database access, to retrieve task information from project database. (Component 80, FIG. 2).

'Remind' module, to send auto reminders to task leaders of pending tasks. (Component 90, FIG. 2).

'Inform' module, to inform task leaders on completion of dependent activities. (Component 100, FIG. 2)

Project reporting, to generate and send out project progress reports. (Component 110, FIG. 2)

The server software system has been implemented in C language, making it easier to port to different machines and also providing the necessary performance for handling the largest projects. The server system has to work closely with the Messaging system and the Network Operating System, the components of which are described below.

FIG. 1, Part 30—Messaging system is used to refer to the communication protocol used to allow users communicate with the 'Auto Multi-Project Server'. This has to work under and is closely related to FIG. 1, Part 40 which is the Network Operating System. The network operating system controls the network operation.

The 'Auto Multi-Project Server' has to be configured separately for each type of messaging system/network operating system. The following are some of the example environments, they are provided for clarity and do not exclude other environments:

UNIX mail system running under a UNIX server machine such as a 486 PC, PENTIUM PC, SUN Workstation, DEC workstation, IBM RS6000 workstation etc. and communicating to other machines via an electronic network (Part 40). Users send electronic mail to the server machine on new project information and status updates, which is processed by the 'Auto Multi-Project Server'. The 'Auto Multi-Project Server' in turn sends back electronic mail reminders and management reports.

Mail system running on a network of PCs'. The mail system may be LOTUS CCMail or LOTUS NOTES or MICROSOFT MAIL etc. The Network Operating System may be NOVELL NETWARE or VINES or PC-NFS etc.

Fax based communication system, based on fax board inside PC coupled with OCR software to read fax messages. Here the fax board and software functions as the messaging system, as well as the network operating system. The fax system must have automatic transmission, automatic receipt and optical character recognition capability.

FIG. 1, Part 50 is the communication network, which may be Fax, LAN (local area network), WAN (wide area network) or telephone network is the physical media for communication. This is generally transparent for the 'Auto Multi-Project Server', as it is handled by the Network Operating System (Part 40).

FIG. 1, Part 60 are the different Project Work-team members communicating to the 'Auto Multi-Project Server', these may be program managers, project leaders, task leaders etc. The program manager assigns inter-project

priorities, the project leader creates the first plan for the project and the task leaders update the project database with progress of project.

Operation of Invention

The 'Auto Multi-Project Server' is designed to change the organization process of planning, tracking and management of work-team projects. The entire process is planned to be automated with managers focusing only on the exceptions and crisis situations. I describe below the process of using the 'Auto Multi-Project Server', as well as the logic used by the Invention to manage projects.

Step 1—Set up the environment, this involves creating an environment configuration file with an entry in it to show the command string to be used in submitting an input file for electronic mail or for faxing it out. Also the command string to be used in reading an electronic mail or fax, and the input file name. The environment file is also used to indicate week-ends, holidays and default reminder frequency and default reminder window.

In addition a separate file containing authorization information is required for 'project leaders to allow them to create new projects. The authorization will consist of a list of 'user names' and valid passwords'.

The above environment configuration files are the only manual operation and is a one time operation. These can be performed by a Systems Manager, who does not require to have any knowledge of 'Project management processes'. Based on the configuration values the appropriate communication procedures are invoked.

Step 2—The project leader creates a project plan in the format as per FIG. 3. This has information on the project, tasks, dependencies and resources to be used.

Step 3—The 'Auto Multi-Project Server' reads the project plan and invokes the 'Build' Procedure as per flowchart in FIG. 4. The first step is to validate the user/password and then parse the input for errors such as if names, descriptions, mail ids etc. are missing. Based on dependencies, the server computes the completion dates for each task. Completion dates are based on work-days only, so the software accounts for week-ends and holidays. Refer FIG. 5—Dependency Computation Flowchart for details on how dependencies are handled.

Step 4—Resolving Inter-project resource conflicts. The 'build' procedure is not complete unless critical common resource usage has been compared against the resource limits. Refer FIG. 9 for the flowchart for the re-allocation of critical resources and the re-computing of completion dates.

Prior to this the Program Manager is mailed a list of projects and is requested to assign a rank priority. In addition the program manager is requested to supply a list of critical resources and their usage limits. The actual usage is compared against this limit.

Step 5—Task Reminders are accompanied by information about the task, to allow task leaders to provide change updates such as changes in the task duration or the estimated completion date or to inform on amount of resources actually consumed. Refer FIG. 8 for the procedure flowchart on updating the project database.

Step 6—The 'Inform' procedure is invoked each time there is a change in a dependent task status such as starting or completion. Refer FIG. 7 for the flowchart. Thus the 'Auto Multi-Project Server' serves also as a formal mode of communication of task status, and does not leave it to be communicated via other informal means.

Step 7—The 'Reports' procedure shown in FIG. 10, is invoked to generate the necessary management reports on time/cost overruns and critical path analysis. In addition managers are informed on re-allocation of any resources, to explain re-computation of completion dates.

Automatic Operation—It should be noted that the operations of the 'Auto Multi-Project Server' are automatic (or self-running) and there is no manual intervention. Only at the staff-up stage is there any manual customization of the environment, by changing some of the program variables and setting up the environment file. In general the operation is driven by mail messages received from users. The 'Auto Multi-Project Server' responds to the messages and is thus 'event driven', though the response does not occur immediately but rather batched together for the end of day processing, as explained in the next paragraph.

Night processing—It is suggested that given the heavy amount of processing for each project, all 'build', 'reminder', 'inform', 'reports', 'resource re-allocation' procedures be run at night. This will allow for lowered load on the server and also allow accumulation of all status updates to be handled together eliminating database file access problems. Real time response to user messages is not necessary.

Summary, Ramifications and Scope

I have described so far the architecture and working of an 'Auto Multi-Project Server' which is necessary to implement the proposed 'Automatic Multiple Project Management Process'. As can be seen from the product description and operation so far, the 'Auto Multi-Project Server' is a first of its kind product with some very unique features to handle large and multiple number of projects, including where there are resource conflicts. The future organizations will be greatly demanding in terms of number and size of projects and only an automated software system can prove to be useful. Existing static tools will be overwhelmed by the amount of project data.

What is claimed as the invention is:

1. A method executed by a computer system as part of a computer program, said system for coordinating the management of a project, said computer system to comprise of a central database server connected to a electronic network, said method using a two way electronic messaging system that allows different types of organizational work-group team members to send messages to the computer program and receive messages from the computer program via the said electronic network, said method storing and accessing data from a multi-project database, said method to be automatic in nature and with built in triggers which are based on the nature and status of said data without need for manual project management coordination, said project management coordination to involve all the steps of the project management cycle including planning, resource leveling, status reporting and reminding, tracking and updating plans, said method to be configurable for the said organizational work-group environment, said method comprising the steps of:

- (a) identifying the owner of received message;
- (b) identifying the nature of said received message;
- (c) setting up said multi-project database and saving said received messages to said database according to the nature of said received message;
- (d) receiving project plans and compiling project plans and saving project plans into said multi-project database;

(e) checking said project plans for resource requests against resource availability and reallocating resources if necessary based on inter-project priorities;

(f) recalculating and sending back said project plans based on resource reallocation;

(g) sending project status reports and reminders to organizational work-group team members based on the status of said triggers;

(h) receiving project updates and status changes and updating said project database;

(i) repeating steps (a) through (h) on a periodic basis, as desired by said organizational work-group team members.

2. A method as claimed in claim 1 wherein said identifying the owner of the message step additionally comprises the step of identifying the incoming user messages and authorizing them against a valid list of users.

3. A method as claimed in claim 1 wherein said identifying the nature of the message step additionally comprises the step of checking the consistency of data sent to it via user messages, prior to processing the messages; has means to reject and return user messages which are inconsistent.

4. A method as claimed in claim 1 wherein said compiling project plans step additionally comprises the step of computing task completion dates based on project start date and inter-dependencies among tasks.

5. A method as claimed in claim 1 wherein said saving project plans step additionally comprises the step of creating a project database of information on the project, with all necessary information for follow-up of progress and all necessary information for reporting progress.

6. A method as claimed in claim 1 wherein said recalculating project plans based on resource reallocation step additionally comprises the step of recomputing task duration's and completion dates of tasks.

7. A method as claimed in claim 1 wherein said sending project status reports and reminders step additionally comprises the step of processing each project database and generating outgoing reminder messages for task leaders whose tasks are pending.

8. A method as claimed in claim 1 wherein said sending project status reports step additionally comprises the step for soliciting for updates to project and database status information.

9. A method as claimed in claim 1 wherein said sending project status reports step additionally comprises the step of sending messages to task leaders, on status of inter-dependent tasks.

* * * * *



US005765140A

United States Patent [19]

Knudson et al.

[11] Patent Number: **5,765,140**[45] Date of Patent: **Jun. 9, 1998**[54] **DYNAMIC PROJECT MANAGEMENT SYSTEM**

[75] Inventors: **James George Knudson**, Monument;
William Lawrence Vivian, Colorado
 Springs, both of Colo.; **Mark S. Crego**,
 Springfield, Va.

[73] Assignee: **MCI Corporation**, Washington, D.C.

[21] Appl. No.: **559,970**

[22] Filed: **Nov. 17, 1995**

[51] Int. Cl.⁶ **G06F 17/60**

[52] U.S. Cl. **705/9**

[58] Field of Search **395/209, 211,**
395/201, 207, 208, 232, 926; 705/8, 9

[56] **References Cited****U.S. PATENT DOCUMENTS**

4,819,162	4/1989	Webb, Jr. et al.	395/232
4,937,743	6/1990	Rassman et al.	398/208
4,942,527	7/1990	Schumacher	395/209
5,111,391	5/1992	Fields et al.	395/209
5,117,353	5/1992	Stipanovich et al.	395/211
5,172,313	12/1992	Schumacher	395/211
5,459,657	10/1995	Wynn et al.	395/232
5,557,515	9/1996	Abbruzzese et al.	395/209

OTHER PUBLICATIONS

MCI, "TES Users' Guide-Manager & Budget Coordinator Edition," revised Aug. 1994, pp.: 1-35.

MCI, "TES Users' Guide-Employee & Contractor Edition," revised Aug. 1994 pp.: 1-23.

MCI, "AIMS Users' Guide," revised Aug. 1994, pp.: 1-24.

Primary Examiner—Donald E. McElheny, Jr.

[57] **ABSTRACT**

A dynamic project management system includes a server network and a master database. The network is configured to identify a personnel resource pool including a plurality of users. A project planning tool is used to effect the project plan including a plurality of tasks to be performed by the users in accordance with respective time schedules. The network is configured for translating the project plan into the master database to effect an assignments table including a list of project tasks assigned for completion by each of the users. Time sheets are periodically prepared in the master database from the assignments table and include a list of the project tasks assigned to a respective user and a time period record for recording time entries therein. Actual time expended in performing the tasks is fed back to the project plan for managing completion of the tasks in accordance with the time schedules. In a preferred embodiment, a funding source is mapped to the project plan so that cumulative labor cost may be tracked based on actual time expended for the project tasks.

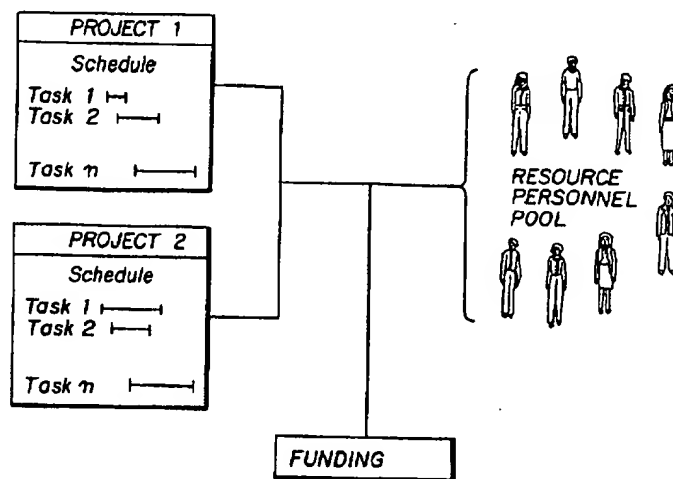
16 Claims, 3 Drawing Sheets

FIG. 1

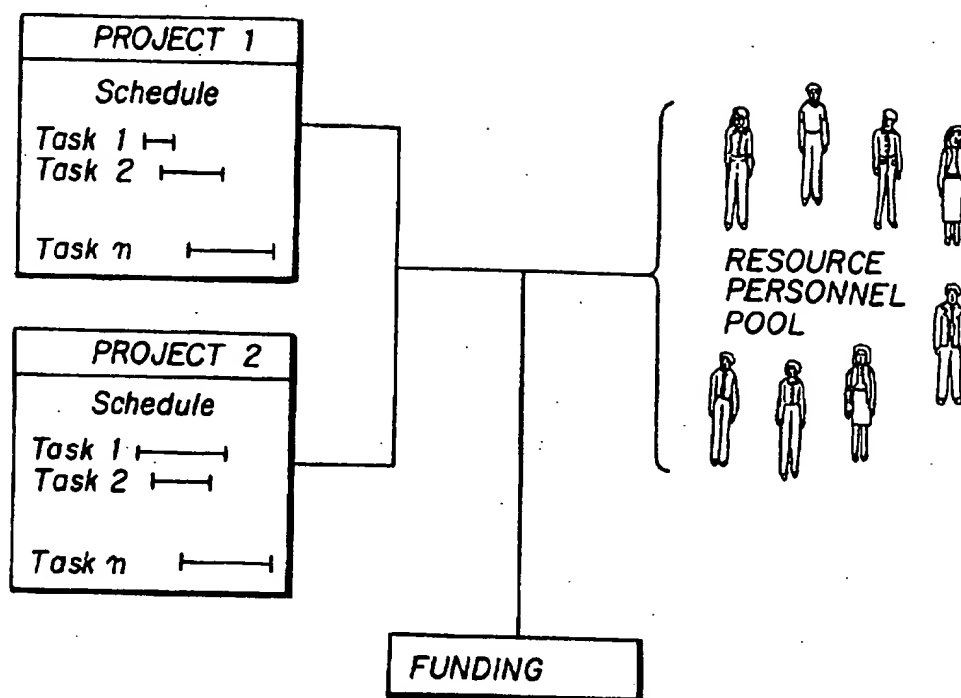


FIG. 2

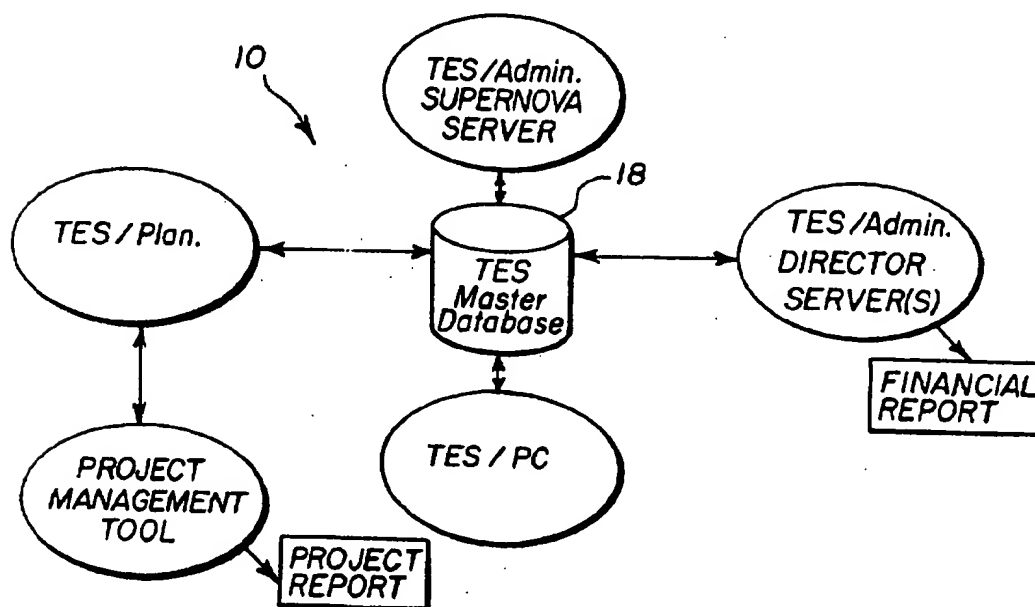


FIG. 3

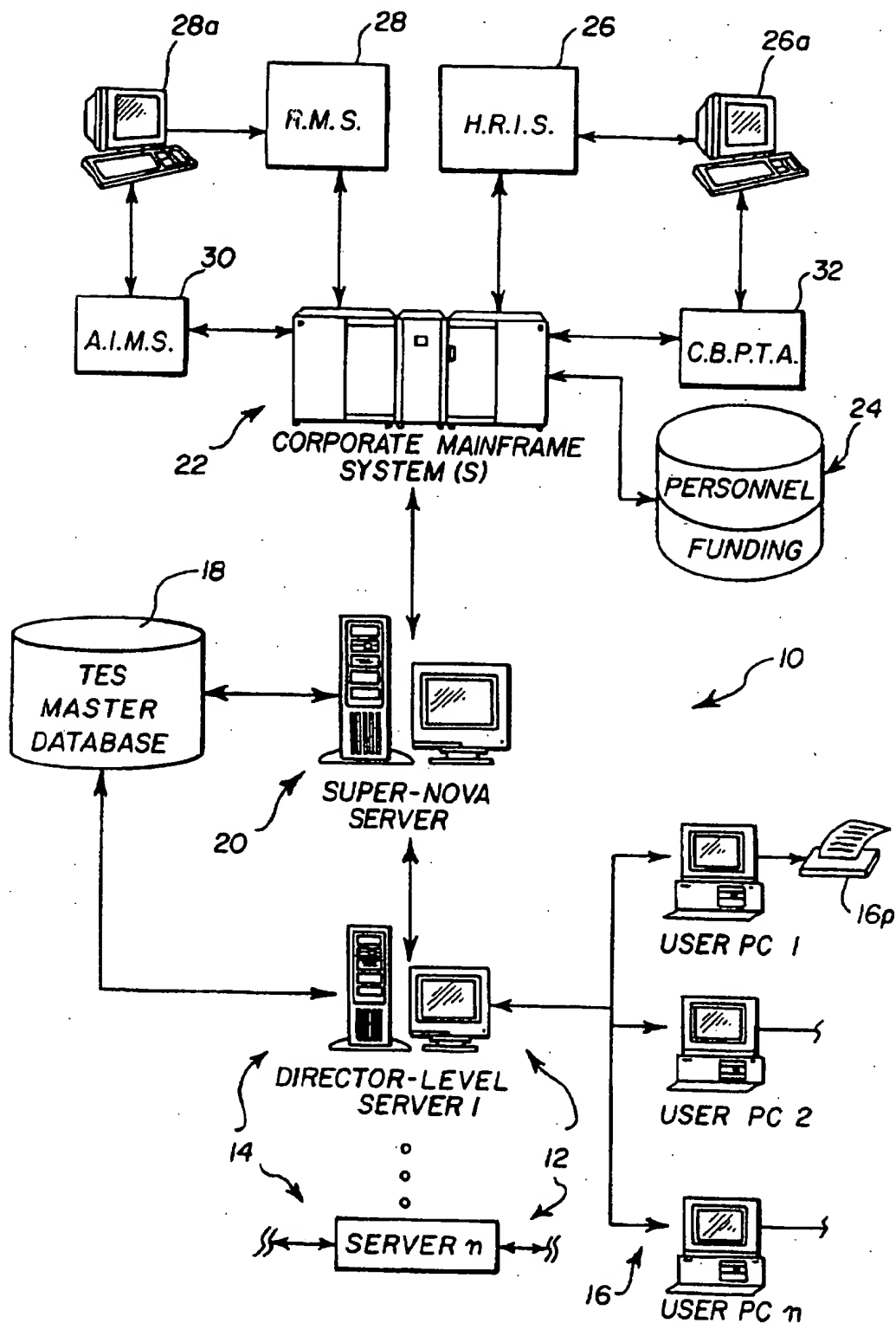
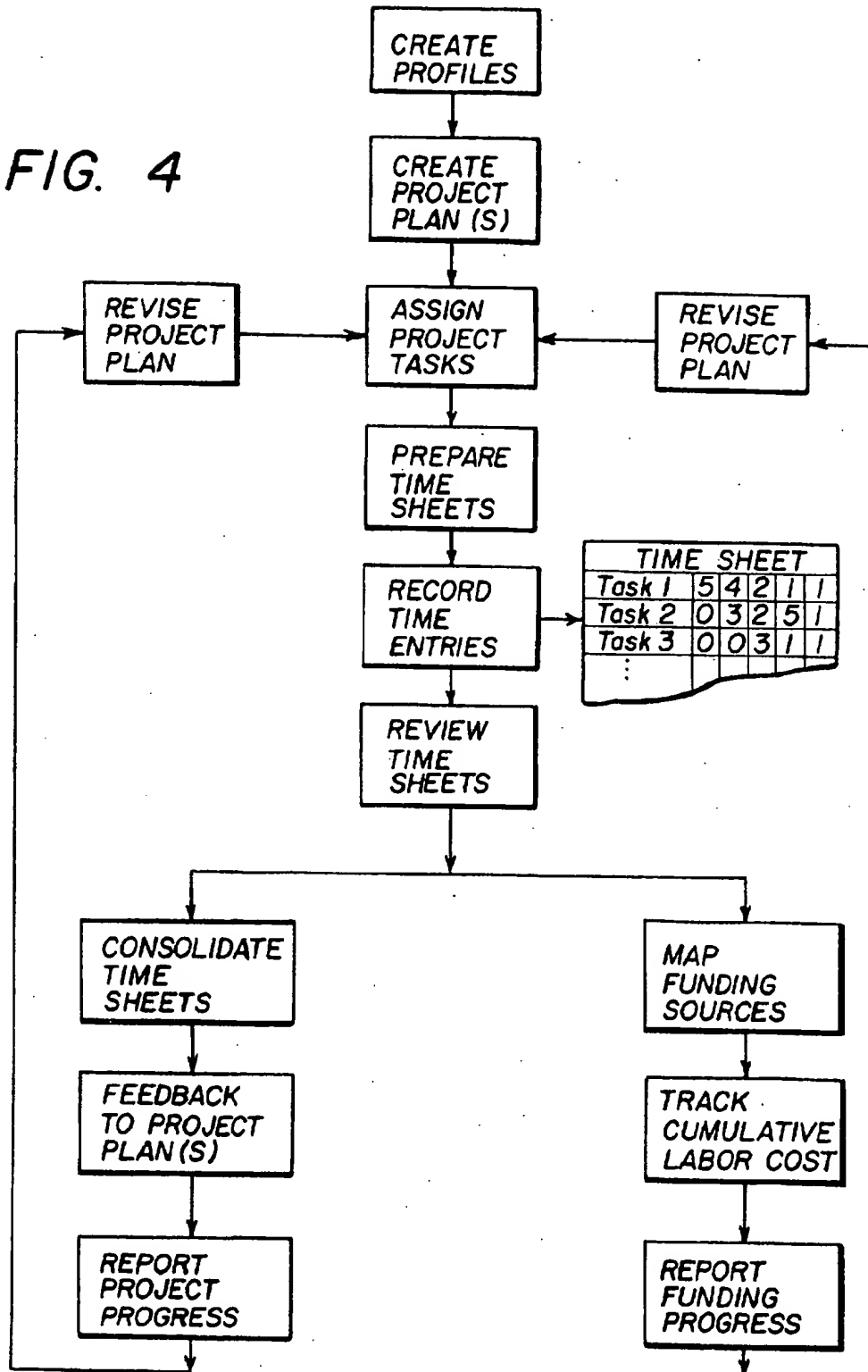


FIG. 4



DYNAMIC PROJECT MANAGEMENT SYSTEM

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates to data processing systems. More particularly, the invention relates to management applications used in data processing systems.

2. Background Information

Large companies employ numerous employees and external contractors for performing various projects in conducting business. Various levels of managers define projects to be performed by employee and contractor personnel in accordance with project plans containing specified tasks requiring completion in accordance with respective time schedules.

Large companies typically utilize programmable computers and associated databases to more effectively manage available personnel and track expended labor hours for following established financial budgets. Various internal and external funding accounts may exist to which personnel may charge their labor time. In one common example, it is required to track work performed in completing a capital project which can be depreciated for tax purposes. Accordingly, a specific Capital Project Appropriation Request (CPAR) is created to which personnel may charge expended time in performing work thereunder.

One conventional system is known as the Automated Issue Management System (AIMS) and is used in conjunction with another conventional system called Entry System (TES) which are used for capitalizing internal and external labor. The systems are contained in conventional programmable computers for more accurately managing the large number of personnel involved and the substantial number of individual tasks associated with various capital projects. A project manager typically initiates a project by opening a specific work request in the AIMS system. The system automatically provides a respective AIMS number, which typically has a correlation to a single CPAR number. The manager assigns personnel to complete tasks based on an AIMS number. The TES system allows personnel to record through a personal computer expended time per AIMS number as the work is completed. The TES system allows the manager to examine cumulative time expended for specific AIMS numbers, with the manager manually correlating the AIMS number to a specific CPAR.

Although the AIMS and TES systems facilitate managing projects in a large company, project planning and managing is nevertheless relatively complex in most circumstances. To further facilitate project management in large companies, commercially available software has been developed. One available management tool or product is conventionally known as Microsoft Project from Microsoft Corp., and another product is ABT Project Workbench from the Applied Business Technology Corporation. These software tools allow companies to define project plans in accordance with tasks and time schedules for available personnel resources, and are typically operated in stand-alone fashion or in conjunction with other commercially available software products for facilitating the overall management of projects. However, these commercially available products are designed for specific applications and are therefore limited in capability, as well as having no built in capability for managing project funding such as the CPAR example introduced above.

SUMMARY OF THE INVENTION

A dynamic project management system includes a server network and a master database. The network is configured to

identify a personnel resource pool including a plurality of users. A project planning tool is used to effect the project plan including a plurality of tasks to be performed by the users in accordance with respective time schedules. The network is configured for translating the project plan into the master database to effect an assignments table including a list of project tasks assigned for completion by each of the users. Time sheets are periodically prepared in the master database from the assignments table and include a list of the project tasks assigned to a respective user and a time period record for recording time entries therein. Actual time expended in performing the tasks is fed back to the project plan for managing completion of the tasks in accordance with the time schedules. In a preferred embodiment, a funding source is mapped to the project plan so that cumulative labor cost may be tracked based on actual time expended for the project tasks.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention, in accordance with preferred and exemplary embodiments, together with further objects and advantages thereof, is more particularly described in the following detailed description taken in conjunction with the accompanying drawings in which:

FIG. 1 is a schematic representation of a plurality of projects to be completed by a personnel resource pool using various funding sources.

FIG. 2 is a schematic representation of a dynamic project management system having a master database in accordance with one architecture embodiment of the present invention.

FIG. 3 is a schematic representation of one architecture embodiment for implementing the dynamic project management system illustrated in FIG. 2.

FIG. 4 is a flow chart illustrating exemplary method steps in effecting the dynamic project management system illustrated in FIGS. 2 and 3.

DESCRIPTION OF THE PREFERRED EMBODIMENT(S)

Illustrated schematically in FIG. 1 are a plurality of project plans, labeled projects 1 and 2 for example, each of which includes a plurality of project tasks which require completion in accordance with respective time schedules as is conventionally known. The various project tasks must be completed by available employees in a typical large company, and by external contractors as required. The employees and contractors define a resource personnel pool from which managers assign one or more tasks to selected individuals from the various projects which must be completed in accordance with respective time schedules. Associated with each of the projects is a respective internal or external source of funding to which each of the project tasks may be assigned.

In accordance with the present invention as illustrated schematically in FIG. 2, a dynamic project management system 10 is provided for automatically tracking and controlling project tasks in accordance with the various project time schedules, and the associated funding therefore. FIG. 3 illustrates schematically the architecture or hardware associated with the system 10, with FIG. 2 illustrating schematically the software associated therewith.

Referring firstly to FIG. 3, the system 10 includes one or more conventional server networks 12 each including a director-level programmable digital computer server 14 operatively joined to a respective plurality of programmable

personal computer (PC) user terminals 16. The number of user terminals 16 is typically limited, to about 250 terminals for example, for each director server 14, and therefore additional server networks 12 are utilized as required for supporting the specific number of resource personnel available, who are also referred to as users. The director servers 14 may be located at one geographic site, or may be located at various geographically spaced apart sites depending upon the organizational structure of the corresponding company and its needs. Conventional software is used to operate the server networks 12.

The one or more director servers 14, and their associated user terminals 16, are conventionally operatively joined to a common or master database 18 to which data may be written and stored, and from which data may be read by each of the director servers 14 and the various user terminals 16. The master database 18 may reside at any one or all of the director servers 14, and may be defined redundantly at each of the servers 14 if desired.

In the exemplary embodiment illustrated in FIG. 3, the system 10 further includes a supernova-level programmable computer server 20, also referred to simply as a superserver 20, operatively joined to the plurality of director servers 14 and the master database 18. The superserver 20 is conventionally configured and operatively joined to the director servers 14 for suitably downloading and uploading data as described in more detail below.

The system 10 preferably includes in part and cooperates with a pre-existing, conventional programmable mainframe computer system 22 which is operatively joined to the superserver 20. The mainframe system 22 includes a main database 24 having stored therein personnel resource data and funding data. In one company for example, the mainframe system 22 would be employed for maintaining suitable records of available company employees, and outside or external contractors typically used thereby as required, with these workers or users being collectively referred to as personnel. Similarly, the mainframe system 22 would also be used for storing suitable identification of various funding sources, both internal and external to the company, as desired so that work tasks and non-work related charges such as vacation and sick days may be associated with a specific funding identification or source.

In one example, the mainframe system 22 conventionally includes a Human Resources Information System (HRIS) 26, which is software implemented and accessible by a conventional terminal 26a. The HRIS system 26 contains, among other things, various data regarding company employees.

The exemplary mainframe system 22 also includes a software implemented Resource Management System (RMS) 28 which is accessed through a corresponding terminal 28a. The RMS system 28 is used to maintain employee and contractor information required for monitoring utilization thereof and cost of development, maintenance, and support of software, for example.

The mainframe system 22 further includes another software implemented conventional Automated Issue Management System (AIMS) 30 which is used for capitalizing internal and external labor. In the AIMS system 30, a manager first opens a work request to begin a Capital project. The AIMS system 30 provides a corresponding AIMS number therefore, which number has a correlation to a single specific funding source such as a CPAR number. The AIMS system 30 is used in conjunction with the conventional time entry system introduced above in which

personnel enter expended time in a computer terminal based on the AIMS number itself.

The mainframe system 22 also includes another conventional, software implemented Capital Budget Project Tracking Analysis (CBPTA) system 32 which is used for tracking Capital budgets independently of the AIMS system 30.

In order to effect the Dynamic Project Management System 10 illustrated in FIG. 3, it is desirable to use available or pre-existing hardware and software to advantage for automatically tracking and controlling funding progress and time schedules for various projects. In this regard, the hardware illustrated in FIG. 3 including the master database 18 is used in conjunction with software developed specifically therefore in accordance with the present invention. Although the conventional time entry system software introduced above facilitates time entry by the various users, such time entry is related to the simple AIMS numbers without any direct correlation to either project funding sources or project plans. In accordance with the present invention, the Dynamic Project Management System 10 includes an improved Time Entry System (TES) configured for specifically associating time tracking with separately developed project plans, as well as funding information.

As illustrated in FIG. 2, the TES software in accordance with the present invention includes three portions or modules referred to separately as TES Administrator (TES/Admin), TES planning (TES/Plan), and TES Personal Computer (TES/PC). Each of these three modules respectively addresses the specific need of personnel resource management, project planning, and time entry in an inter-related cooperation using the common or master TES database 18. The TES/Admin software resides in the exemplary embodiment illustrated in FIG. 2 on both the director servers 14 and the supernova server 20, with different functions being performed on each server as described in more detail below. The TES/PC software resides at the user terminals 16 which are used primarily for entering expended time into the master database 18. This setup allows users to act independently, with the common database 18 being readily maintained independently thereof. And, TES/Plan also resides on the director servers 14 for providing a specifically configured interface with a conventional, commercially available project management tool in accordance with the invention.

More specifically, two examples of commercially available project management or planning software tools include Microsoft Project from the Microsoft Corporation, and ABT Project Workbench by the Applied Business Technology Corporation. Each of these preexisting and predefined software tools provides means for a project manager to plan a desired project to effect a project plan including a plurality of described tasks to be performed by identified personnel or users in accordance with respective time schedules. The TES/Plan module in accordance with the invention as illustrated in FIG. 2 provides means for automatically interfacing the existing project management tool with the TES server network 12 through the common database 18. In order to maximize compatibility between the preexisting project management tool and the TES server network 12, the TES software modules including TES/Plan, TES/Admin, and TES/PC are created using the commercially available Microsoft Visual Basic software which allows users to interface with the system at their respective terminals using "windows" analogous to those found in the well known Microsoft Windows software.

5

Since the project management tool creates project tasks and assigns them to identified users, it is desirable to correlate identification of the users in the project tool to actual users in the Project Management System 10 already identified therein. More specifically, the mainframe system 22 previously described with respect to FIG. 3 includes the main database 24 which identifies available employee and contractor personnel. Accordingly, the TES/Admin software module which resides on the superserver 20 illustrated in FIG. 3 is used as a resource administrator for the TES applications and is used in conjunction with the existing mainframe system 22 as means for providing or identifying a personnel resource pool from the personnel data stored in the main database 24.

TES/Admin is used to provide each of the personnel or users with a corresponding identification profile which is used to advantage in the TES system including the project management tool. Each user profile includes a unique identifier, such as a four character identification (ID), that identifies each user in the system 10 and controls access thereto. The user profile also includes the user's name and a password which may be selected by the user; a reporting-to manager; a time sheet approving manager; and a security or access level flag, for example for levels 1 through 9, which determines permitted access to various functions of the TES system 10. The user profile also includes a directorate, or department and location, identification which allows the user profiles to be organized to correspond with respective ones of the director servers 14, with respective director servers 14 being used locally to service or administer each of the users assigned within its jurisdiction.

Since the superserver 20 is operatively joined to the mainframe system 22, the TES/Admin module at the superserver level is configured to effect the personnel resource pool by downloading employee information from the HRIS system 26 and may be used to manage employee transfers, additions, terminations, and associated information. Contractor profiles are maintained by downloading data from the RMS system 28 in a similar manner, with the respective personnel data being retrieved from the main database 24. The personnel profiles are organized by TES/Admin by directorate which is associated with a respective one of the director servers 14.

And, most significantly, the available personnel resource pool as identified by their respective profiles is also available for use in each of the director servers 14 by project managers for use in developing required project plans using the project management tool shown in FIG. 2. The project management tool may reside at any one or all of the director servers 14, and preferably uses a personnel resource template therein identifying users selected by project managers using TES/Admin from the available personnel resource pool maintained in the master database 18. The TES/Plan interface module illustrated in FIG. 2 downloads to the project management tool identified users from the personnel resource pool for use in assigning tasks thereto in accordance with the project plan separately developed by the project management tool.

Any project manager can create one or more project plans using the project management tool as accessed through any user terminal 16, with the manager's own user profile being associated therewith for in effect owning or controlling the respective project plans. These first two steps of the overall method are illustrated in flow chart form in FIG. 4 to initially create the user profiles using the TES/Admin module, and then creating one or more project plans using the commercially available project management tool based on the so

6

created user profiles. Project managers may also use TES/Admin to define general tasks to be completed by selected users independently of the project management tool.

The TES/Plan interface module is also configured for translating the project plan developed in the project management tool into the TES master database 18 illustrated in FIGS. 2 and 3 to assign project tasks to individual users by effecting an assignments table in the master database 18 which includes a list of project tasks assigned for completion by each of the identified or assigned users. The assignments table resides in the common or master database 18 and is accessible by each of the director servers 14 and lists assigned tasks for one or more projects for each of the identified users.

The TES/Plan interface module extracts the tasks and assignments from the project management tool using one of two exemplary methods. For the Microsoft Project tool, conventional Object Linking Embedded (OLE) is used. The tasks and assignments are saved in the TES master database 18 exactly in the same format as the Microsoft Access database created by Microsoft Project which makes project loading and project updating extremely easy to perform in the TES software. For the ABT Project Workbench, the ABT project file is saved by the user manager in a conventional comma-delimited format. The TES/Plan module may then update directly this comma-delimited file. In all cases, the four character unique user identifier serves as the key database field to interface the project management tool with the remainder of the TES system 10.

Referring again to FIG. 4, after the assignments table is created in the master database 18 for assigning project tasks to users identified by their user profiles, the TES/PC software module illustrated in FIG. 2 is then used by the users for manually entering actual or expended time in accomplishing the project tasks using a visual or virtual time sheet.

More specifically, the TES/PC software module which resides at each of the user terminals 16 illustrated in FIG. 3 and allows access by authorized users to the common master database 18 illustrated in FIGS. 2 and 3. The TES/PC module provides means for periodically, on a weekly schedule for example, preparing in the master database 18 from the assignments table therein a plurality of virtual time sheets, one for each respective user, with each time sheet listing the several project tasks assigned to a respective user and a time period record for manually recording time entries by the users indicative of actual time expended by each user in performing the respective task. As shown in FIG. 4, the time sheets are automatically prepared for each user and provide a simple vehicle for the user to enter actual time expended against identified and described project tasks, which is a significant advantage over the conventional time entry system which allows time entry based on the simple AIMS number.

A schematic representation of a portion of an exemplary time sheet is illustrated in FIG. 4 for a given time period such as an identified week during which listed tasks may be completed in accordance with the time schedules established in the project management tool. Since the TES/PC software module is also written in Microsoft Visual Basic software, it will have the typical Microsoft "windows" appearance although being configured to suitably identify individual users, and list their assigned project tasks by name and/or description, and provide suitable table records in which expended time may be recorded against assigned tasks. The time sheet preferably also includes a comments region in which the user may provide any suitable comments directed

to the approving manager. The TES/PC module may be suitably configured as desired for displaying one or more suitable windows for informing each user of assigned tasks and time schedules therefore, while allowing the user to enter actual time expended in performing the tasks, as well as estimated time to completion for the tasks. Various other categories may also be presented in the windows which are typically used in conventional time sheets including recording vacation, illness, or personal business days for example. Of primary significance, however, is the ability for users to enter labor expended against specific tasks, instead of entering time on simple AIMS numbers which will allow project teams to track and control projects with improved efficiency and accuracy.

After a user completes a time sheet, it is stored in the master database 18 and may be printed by the user at any time using a conventional printer 16p as illustrated in FIG. 3. Each user is assigned a time approving manager who has security access to the time sheet data and is required to review the time sheets and either approve or reject the entries made therein. If rejected, the user is suitably alerted via the network so that any required corrections may be made thereto. Once time sheets are approved, the data contained therein are then available for further analysis and use in the TES system 10.

For example, and as shown in FIG. 4, the TES/Plan module is also configured to organize and consolidate the time sheet entries as desired for individual projects and tasks therein, which data is then fed back to the project plans in the project management tool. This provides the project manager an accurate feedback loop on actual progress made against a specific project plan. Suitable project progress reports may be configured for examination by the project manager either visually on his user terminal or may be printed as desired for managing completion of the project tasks in accordance with the various time schedules. The actual time expended by the user is simply fed back into the assignments table maintained in the master database 18 in the same format as required for access by the project management tool. In this way, project managers may periodically track and control project progress in accordance with the previously defined time schedules. If required, the project management tool may be used to revise the various project plans and in turn update the assignments table maintained in the master database 18 to adjust assigned tasks and time schedules as required for the various identified users.

The TES/Admin module residing at the superserver illustrated in FIG. 2 also provides means for administering the personnel resource pool to effect changes thereto and in turn changes to the assignments table maintained in the master database 18. The personnel resource pool may be modified in response to transfers, additions, and terminations of users, as well as changing user profiles as required. For example, user passwords may be reset as desired and changes of directorate and reporting or approving managers may also be changed. Security level access to the TES system 10 may be changed as required by modifying the corresponding security level flag or parameter. The lowest level security is that for the individual users for allowing their access to the user terminals 16 for entering time therein. Reporting and department managers have a correspondingly higher security level for allowing their access to the TES system 10 as required for completing their responsibilities. Project managers are given suitable security level access to load projects into TES/Plan which are created in the project management tool and develop or revise project plans. The highest level of

security resides at the superserver level for administering all aspects of the TES system 10 including the personnel resource pool.

Another significant feature of the present invention is the ability of the TES/Admin module to map or assign a specific funding source to a specific project or project plan so that cumulative labor costs therefore may be tracked based on actual time expended for the project tasks by the assigned users.

As illustrated in FIG. 4, the TES/Admin module is suitably configured to map a desired funding source to specific projects and project tasks therein identified in the individual time sheets and stored in the assignments table in the master database 18. As illustrated in FIG. 3, existing funding sources are identified in the main database 24 of the mainframe system 22 which may be downloaded as desired for association with respective project plans identified in the master database 18. A specific funding source may be readily linked to a specific project plan in the master database 18 so that cumulative labor cost therefore may be tracked. The funding sources may either be internal to the company, or external such as joint venture credits and alliance reporting. Also, temporary funding sources may be generated in TES/Admin that can later be mapped to valid corporate accounts if desired. A typical corporate funding source is the conventional CPAR. And, funding sources for specific project plans may be changed periodically as desired using the TES/Admin module.

As illustrated in FIG. 4, tracking cumulative labor costs for the individual project plans may then be used for reporting funding progress for the plans, which allows a project manager to revise project plans as required in the project management tool and correspondingly revise the assignments table in turn. In this way, the TES system 10 allows tracking of not only project progress in accordance with the time schedules, but also tracking of cumulative labor cost as actually expended by users in performing the project tasks. The project plans may therefore be suitably revised as required to accommodate time schedule and funding concerns.

Although the TES/Admin software module and the TES/Plan module may be maintained on a single director level server 14, it is desirable to create the multi-level hierarchy architecture illustrated in FIG. 3 for integrating the TES modules with the mainframe system 22 for improving performance. For example, the TES/Admin module residing at the superserver 20 may be effectively used for identifying all director server locations; identifying all authorized users and their respective server locations; feeding all applicable data to the director servers 14; and consolidating and backing up data from the director servers 14. The superserver 20 provides a repository for consolidated project data from all downstream director servers 14, and performs all required interface functions with the corporate mainframe systems 22.

Through downloads from the corporate system 22, the TES/Admin module identifies available users based on active employees and contractors from the personnel data residing at the main database 24. It also validates the funding information for the various project plans based on the corporate accounting systems residing at the corporate level. Upgrades to the various TES software modules may be also downloaded as required.

Through uploads to the corporate mainframe system 22, the TES/Admin module provides managed labor feed to the appropriate corporate time accounting systems, including all

applicable required edits. Conventional corporate time accounting requirements are expanded to include functionality for billing and charge back of labor to external alliances and joint ventures.

The TES/Admin module residing at the director level preferably provides at this level the ability to set user parameters including for example, user access to all TES software; passwords; approving managers and subordinate roll up structure; and user roles. User role definitions for TES/Admin use include identification of approving managers, without budget authorization; department managers, with complete authorization for selected departments; project managers, having complete authorization for selected projects; and director level administrators having complete authorization for corresponding functions. General tasks independent of the project management tool may be created by responsible managers at the director level for completion by assigned users using a generic time funding category.

The TES/Admin module is preferably also used to define funding relationships for the project plans loaded through the TES/Plan module by linking project plans to corporate funding sources, which may be changed periodically as desired. Temporary funding sources may also be assigned, and direct links to the corporate mainframe system 22 are preferred for validating funding sources and editing as required. And, any type of funding source may be identified including both internal and external funding.

The TES/Admin module is also configured to prepare various reports regarding performance as required. The commercially available Crystal Reports software available from the Crystal Computer Services Company, Inc., of Vancouver, British Columbia, Canada may be used for example. Complete reporting functionality for both financial and project metrics is provided for tracking project time scheduled progress as well as funding progress as addressed above.

And, the TES/PC module residing at the various user terminals 16 allows the two primary functions of time entry interface for authorized employees and contractors, and subsequent approval by approving managers identified in the TES/Admin module. The users enter time against tasks assigned from the project plans loaded in by the TES/Plan module; or general tasks created in the TES/Admin module for the entire directorate; or administrative accounts for vacation, holiday, sickness, and other typical non-work related conditions. The TES/PC module allows for general as well as task level comments for subsequent status reporting and review by managers. The time sheets are entered periodically such as weekly, with the users being able to enter estimated time to completion against assigned tasks which is fed back into the external project planning tool through the update actuals option in the TES/Plan module. And, the individual users may print copies of the time sheets as desired. Approving managers have security level access to view all subordinate time sheets for approval or rejection thereof. Approving managers themselves are allowed to report delinquencies and administrator account information for all subordinates.

The project management system 10 disclosed above may be readily tailored as desired for integrating the various roles of the users and managers in conjunction with the preexisting project management tool and mainframe system 22. In this way, existing personnel data may be suitably downloaded to the project management tool for creating one or more project plans having tasks assigned to specific, pre-

identified users in accordance with respective time schedules. The various tasks required in the one or more projects are maintained in the common assignments table residing in the master database 18 so that individual users may access the database via the corresponding time sheets for viewing tasks assigned to the users for completion in accordance with the defined time schedules. The individual users readily enter time expended in completing identified tasks, as well as time to completion if desired. The actual time expended is then consolidated in the master database 18 so that project progress in accordance with the various time schedules may be tracked, and most significantly, cumulative labor expended or labor costs may also be tracked. The original project plans may be revised as desired based on the feedback obtained from the project schedule or funding progress. And, of particular importance is the ability to identify project tasks by description or name instead of a non-descriptive AIMS number as was previously done, which allows integration with the project management tool for tracking and controlling identified projects in accordance with the assigned project task.

While there have been described herein what are considered to be preferred and exemplary embodiments of the present invention, other modifications of the invention shall be apparent to those skilled in the art from the teachings herein, and it is, therefore, desired to be secured in the appended claims all such modifications as fall within the true spirit and scope of the invention.

Accordingly, what is desired to be secured by Letters Patent of the United States is the invention as defined and differentiated in the following claims:

We claim:

1. A dynamic project management system comprising:

a server network including a programmable computer server operatively joined to a plurality of personal computer user terminals;

a master database operatively accessible by said server and said user terminals;

means for identifying a personnel resource pool including a plurality of users each having an identification profile;

means for planning a project to effect a project plan including a plurality of tasks to be performed by said users from said resource pool in accordance with respective time schedules;

means for interfacing said project plan with said server network to effect in said master database an assignments table including a list of said project tasks assigned for completion by each of said users;

means for periodically preparing in said master database a plurality of time sheets from said assignments table including a list of said project tasks assigned to a respective user and a time period record for recording time entries indicative of actual time expended by each user in performing said tasks; and

said interfacing means being effective also for feeding back to said planning means from said master database said actual time expended for said tasks for managing completion of said tasks in accordance with said time schedules.

2. A system according to claim 1 further comprising means for administering said personnel resource pool to effect changes thereto and in turn changes to said assignments table.

3. A system according to claim 2 wherein said administering means is effective also for mapping a funding source to said project and tracking cumulative labor cost therefore based on said actual time expended for said project tasks.

11

4. A system according to claim 3 further comprising a plurality of said server networks operatively joined to said master database to define a respective plurality of director-level computer servers each having a respective plurality of said user terminals, and wherein said planning and interfacing means reside at said director-level servers. 5

5. A system according to claim 4 further comprising a supernova-level programmable computer server operatively joined to said plurality of director-level servers and said master database, and including said personnel identifying means. 10

6. A system according to claim 5 further comprising a programmable mainframe computer system operatively joined to said supernova server and including personnel and funding data, and wherein said identifying means effect said personnel resource pool from said personnel data and organize said user profiles to correspond with respective ones of said director servers, and wherein said administering means map said funding source from said funding data to said project. 15

7. A system according to claim 6 wherein said user profile includes a unique identifier, and said planning means are effective to plan said project around users identified by said identifiers. 20

8. A system according to claim 7 wherein said user profile further includes a directorate identification for assigning said users to respective ones of said director servers. 25

9. A method for dynamically managing a project in a server network including a programmable computer server and a master database operatively joined to a plurality of personal computer user terminals, said method comprising: 30

identifying a personnel resource pool including a plurality of users each having identification profile;

planning a project to effect a project plan including a plurality of tasks to be performed by said users from said resource pool in accordance with respective time schedules; 35

translating said project plan into said master database to effect an assignments table including a list of project tasks assigned for completion by each of said users;

12

preparing periodically in said master database time sheets from said assignments table including a list of said project tasks assigned to a respective user and a time period record for recording time entries indicative of actual time expended by each user in performing said tasks; and

feeding back to said project plan said actual time expended in performing said tasks for managing completion of said tasks in according with said time schedules.

10. A method according to claim 9 further comprising: mapping a funding source to said project plan; and tracking cumulative labor cost for said project plan based on said actual time expended for said project tasks.

11. A method according to claim 10 further comprising administering said personnel resource pool to effect changes thereto and in turn changes to said assignments table.

12. A method according to claim 10 further comprising revising said project plan based on at least one of progress in meeting said time schedules and said cumulative labor cost.

13. A method according to claim 10 further comprising planning and translating a plurality of said projects into a common assignments table to effect time sheets for said users containing all project tasks for each user.

14. A method according to claim 10 wherein said project planning step utilizes a preexisting project management tool to effect said project plan.

15. A method according to claim 10 wherein said user profile includes a unique identifier, and further comprising providing said user identifiers to said planning step for planning said project around users identified by said identifier. 35

16. A method according to claim 10 further comprising downloading personnel data from a preexisting computer system for identifying said personnel resource pool.

* * * * *